Template reference variable

- It is used in the html element (any of)
- The symbol is #
- #reference.name
- By giving reference name, it will be refering to the entire element.

app.component.ts

```
import{Component} from '@angular/core';
@Component({
selector: 'app-root',
templateUrl:
styleUrls:
})
export class AppComponent{
   public name=" ";
public logMsg (val)
{ console.log(val)
}
}
```

app.component.html

```
<h1> Welcome to template reference </h1>       → reference variable
<input type="text" #myInput>
<button (click)="logMsg(myInput)"> click </button>
```

Output:
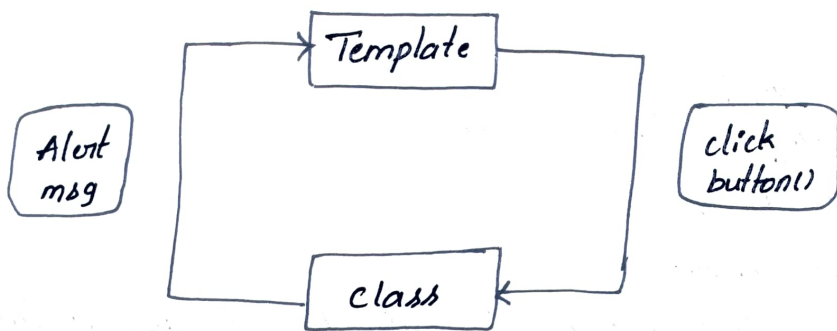
Welcome to template reference variable.

VIRAT	Click

Console Screen:-

```
<input _ngcontent-utb-co type="text">
```

Two Way Binding :-

```
              ┌──────────────┐
        ┌────▶│   Template   │────┐
        │     └──────────────┘    │
 ┌──────┴─┐                    ┌──▼──────┐
 │ Alert  │                    │  click  │
 │  msg   │                    │ button()│
 └──────┬─┘                    └──▲──────┘
        │     ┌──────────────┐    │
        └─────│    class     │◀───┘
              └──────────────┘
```

- Whenever the user enters the data in input tag, in order to fetch the data we make use of value property. That value (from the template to class) is flowing from html page to typescript page. It is because of event binding
- Once the name is getting updated (this.name = val), the name is passed to html page(ts file —> template file)

app.component.ts

```
import { Component} from '@angular/core';
@Component ({
selector: 'app-root'
templateUrl: './app.component.html';
styleUrls: ['./app.component.css']
})
export class AppComponent{
  name = "        ·
  logMsg (val)
  {
    this·name = val;
  }
}
```

Output:-

Welcome to template reference

VIRAT	click

VIRAT

app.component.html

```
<h1> welcome to template reference variable</h1>
<input type = "text"  #inputmsg >
<button (click) =" logMsg ( inputmsg.value)"> click </button>
<h1> {{ name}} </h1>
```

ngModel (Two way Binding)

(event binding)

     + $\Longrightarrow$ [(ngModel)] = Two way Binding.

[property binding]

- The most popular and widely used data binding mechanism is two way binding in angular framework. Basically two way binding is mainly used in the input type or any form element. (where user type or provide any value or change any control value in the one side and on the otherside the same automatically updates in the variable.

- Two way binding is the combination of both property binding and event binding. "( )" represents event & "[ ]" represents binding. whenever we uu use [(ng Model)] in the template page, it has to be imported in app.module.ts

app.module.ts

import { BrowserModule} from '@angular/platform-browser';

:
:

import {FormsModule} from '@angular/forms'

@NgModule ({

  imports: [

    BrowserModule,

    AppRoutingModule,

    FormsModule

  ],

})

export class AppModule

  {

  }

app.component.ts

```
import { Component} from  '@angular/core';
@Component ({
 selector : 'app-root'
 templateUrl:
 styleUrls:
})
export class AppComponent
{
    public name = " ";
}
```

dataBinding []

app.component.html

```
<h1> ngModel </h1>
<Input type = "text"   [(ngModel)] = "name">
<h1>{{name}} </h1>
```

event Binding ()

Output:-

ng Model

Lavanya

Lavanya

Angular Pipes

- Angular pipes can be used to transform data to desired output.
- Pipes takes in a data input and transforms data to a different output.
- Using the pipe operator (1) we can apply the pipes features to any of the property in our Angular.
- Pipes in angular are used to transform the data before displaying it in a browser.

app. component. ts
```
import { Component} from '@angular/core';

@Component ({
  selector: 'app-root';.
  templateUnl:
  styleUrls:
})
export class AppComponent{
  name = "yogesH chandro"
  person = {
    name: "Sachin"
    job: "Cricketer"
  }
  DOB = new Date()
  salary : number = 60000
}
```

app. component. html
```
<h2> {{name}} </h2> // yogesH chandra
<h2> {{name | lower case}}</h2>//yogesh chandra
<h2> {{ name | uppercase}} </h2>//YOGESH CHANDRA
<h2>{{name | titlecase}}</h2>// Yogesh Chandra
<h2> {{name | slice :3}} </h2> // esH Chandra
<h2>{{name|slice :3:6}}</h2> // es H
<h2>{{person |json}} </h2> {"name: "Sachin", "job": "cricketer"
<h2> {{DOB | date:"medium"}} </h2>Wed Apr 08 2020 01:64: 40 GMT+0530
<h2>{{ DOB | date : "short"}}</h2> Apr 8,2020, 1:09:40AM
<h2>{{ DOB | date: "dd/mm/yyyy"}}</h2> 08/09/2020
<h2> {{salary | currency }} </h2> // $60.000
```